Spell Out before You Merge

Hisao Tokizaki

In this paper, I will investigate the relation between the syntactic derivation of sentences and the Spell-Out to PF (Phonetic Form). A problem in the interface between syntax and PF is how computation merges words or constituents and Spells Out them in a bottom-up and cyclic fashion, while PF-representation is linear from left to right (cf. Chomsky 1995, Phillips 2003). The aim of this paper is to show that the problem can be solved if words are Spelled Out before they are Merged. I will argue that lexical items and silent demibeats (cf. Selkirk 1984) are introduced to the working space from top to bottom, and that an extra silent demibeat triggers Merge, which proceeds from bottom to top. This model gives us a new view of the syntax-phonology interface and the architecture of grammar.[1]

1. A Problem: Bottom-Up Merge and Top-Down Linearization

In the minimalist program, it is assumed that lexical items selected from the lexicon are introduced in the working space and are merged with each other step by step. Consider the following sentence for example:

(1) Alice loves small hamsters.

---

[1] This is a revised and enlarged version of a chapter of Tokizaki (2006). I would like to thank Kleanthes Grohmann, Richard Kayne, Howard Lasnik, Satoshi Oku, Kayono Shiobara, Yoshihito Dobashi, and the reviewer of this paper for their valuable comments and suggestions. Thanks go to William Green for correcting stylistic errors. All remaining errors are, of course, my own. This work is supported by Grant-in-Aid for Scientific Research and Sapporo University.

The lexical items needed to derive (1) are {*Alice, Infl, loves, hamsters*}. The derivation proceeds as shown in (2).

(2) a. [small hamsters]
b. [loves [small hamsters]]
c. [Infl [loves [small hamsters]]]
d. [Alice [Infl [loves [small hamsters]]]]

First, the two words *small* and *hamsters* are introduced to the working space and are merged as in (2a). Next, another lexical item *loves* is introduced into the derivation and is merged with *[small hamsters]* as in (2b). The same process applies to *Infl* and *Alice* as shown in (2c) and (2d). Merge proceeds in a bottom-up fashion. Let us call this kind of derivation Merge Up.[2]

From a phonological point of view, PF representation must be linear, running from left to right. PF-representation of the example sentence (1) extends through the following steps:

(3) a. Alice
b. Alice loves
c. Alice loves small
d. Alice loves small hamsters.

Thus, in right-branching structure, the process of linearization starts from the top of a syntactic tree and proceeds down to the bottom of the tree. Let us call this process Linearize Down.

---

[2] I thank a reviewer for the term and comments on the earlier version.

The order of (2a-d) and the order (3a-d) show that there is a tension between syntactic derivation and linearization. Merge proceeds upward while Linearization proceeds downward. Any plausible theories of the syntax-phonology interface must explain this paradox in some way.[3]

Note that the paradox cannot be explained by phase theories of syntax-phonology interface. If we assume that the sister of a strong phase head (*v* and C), namely VP and IP (or TP), is Spelled Out to PF (Chomsky 2001), the sentence (1) is derived as shown in (4).

(4) a. [small hamsters]

b. [loves [small hamsters]]

c. [*v* [$_{VP}$ *loves [small hamsters]*] Spell-Out 1: [$_{VP}$ *loves hamsters]* → PF

d. [Infl [*v* [$_{VP}$ ...]]]

e. [$_{IP}$ Alice [Infl [*v* [$_{VP}$ ...]]] Spell-Out 2: [$_{IP}$ Alice [Infl [*v* [$_{VP}$ ...]]] → PF

First the VP is Spelled Out at the stage (4c), and then the IP is Spelled Out at the stage (4e). If this order is the one in which Linearization takes place, we wrongly predict the following steps in PF-representation resulting in the word order in (5b).

(5) a. loves small hamsters

b. loves small hamsters Alice

---

[3] The paradox becomes apparent when Merge replaces Rewriting rules in syntax. For example, rewriting rules expand VP *loves small hamsters* into V *loves* and NP *small hamsters*. The NP is in turn expanded into *small* and *hamsters*. It is possible to argue that the order of derivation and lexical insertion is top-down. In this sense, the paradox of direction between derivation and linearization is a new problem in the theory of grammar.

To get the right word order in (1), we must assume that the first output of Spell-Out, VP, is stored somewhere in the PF component, and that the second output, IP, is placed at its left.

(6) a. loves small hamsters

b. Alice loves small hamsters

However, as Dobashi (2003) argues, it is not clear how we can guarantee the procedure in the PF component.[4]

The problem of word order is more serious if we consider the derivation and Linearization of sentences with multiple embedded clauses such as (7).

(7) $[_{IP3}$ Mary $[_{VP3}$ thinks that $[_{IP2}$ John $[_{VP2}$ believes that $[_{IP1}$ Alice $[_{VP1}$ loves hamsters]]]]]]

This sentence has the following phase units:

(8) a. $[_{VP1}$ loves small hamsters]

b. $[_{IP1}$ Alice *v* $[_{VP1}$ …*]*]

c. $[_{VP2}$ believes that $[_{IP1}$ …*]*]

d. $[_{IP2}$ John *v* $[_{VP2}$ …*]*]

e. $[_{VP3}$ thinks that $[_{IP2}$ …*]*]

f. $[_{IP3}$ Mary *v* $[_{VP3}$ …*]*]

The result of multiple Spell-Out would be (9), which is a wrong prediction.

---

[4] Dobashi (2003) calls this undecidability of word order "the assembly problem", and proposes that the initial element in the linear string of each Spell-Out should be available to the next Spell-Out. I will not discuss his proposal here.

(9) loves small hamsters Alice believes that John thinks that Mary

This problem becomes even more serious if we assume a hierarchical structure for a discourse. Larson (1990) argues that Coordination structures fall under X-bar theory and have conjunctions as their heads and that discourses are extended coordinations in their default form, as shown in (10).

(10) a. [[He came in] [and [John was tired]]]

b. [[He came in.] [& [John was tired.]]]

Then, Merge can proceed upward infinitely, as shown in (11).

(11) [[$_{IP}$John [$_{VP}$woke up] ... [$_{\&2'}$ $\&_2$ [$_{\&P1}$ [$_{IP}$He$_1$ [$_{VP}$washed his face] [$_{\&1'}$ $\&_1$ [$_{IP}$He$_2$ [$_{VP}$went out]]]]]

Again, multiple Spell-Out cyclically sends the phase units, IPs and VPs, in (11) to make a wrong PF-representation in (12).

(12) went out he$_2$ washed his face he$_1$ … woke up John

It is clear that Linearization must start with the topmost phase unit and then go down to the lowest cycle in order to give the right order in (11). The point here is that a sentence or a discourse can consist of an infinite number of phase units. In the standard assumption in the current syntactic theory, Merge Up starts from the lowest phase unit while Linearize Down starts from the highest phase units.

Note that the same problems occur with other theories of phase such as Uriagereka (1999) and strong derivational theory such as Epstein et al. (1998). They propose different domains of phase unit, but they assume that Merge applies in a bottom-up fashion to build up a sentence. If the phase units are smaller than VPs and IPs, as in strong derivational theory, the problem is more serious. The word order in PF-representation will be almost the opposite of the actual sentence or discourse if each Merge Spells Out the resulting constituent, as shown in (13).

(13) a. [Alice [loves [small hamsters]]]

b. small hamsters loves Alice

We have seen that bottom-up Merge raises a problem with top-down Linearization in the multiple Spell-Out model of grammar. In the following sections, we will consider how we can solve the problem in the minimalist framework.

2. Branch Right and Its Problems

The up-down paradox we have seen above does not occur in the incremental model of derivation proposed by Phillips (1996, 2003), who assumes Branch Right instead of Merge. For example, a VP with Ns as its specifier and complement is constructed in the order shown in (14) (cf. Richards 1999).

(14) a. [Mary saw]

b. [Mary [saw John]]

It is argued that the right node V dominating *saw* in (14a) branches into VP *[saw John]* as shown in (14b). The parser proceeds from left to right, and the structure is built in a top-down fashion at the same time.[5]

This theory avoids the up-down paradox that occurs in the standard minimalist syntax with Merge and Linearization. However, the status of Branch Right in the grammar is not clear. Phillips does not specify how Branch Right applies to a single lexical item and makes it branch into two nodes. In other words, we do not know what mechanism changes the verb *saw* in (14a) into the VP *saw John* in (15b).

Moreover, Branch Right has an empirical problem. As Shiobara (2005) points out, it cannot build left branching structure such as the subject in (15).

(15) [[The girl] [saw John]]

Branch Right wrongly builds *[The [girl [saw John]]]*. This problem is more serious in building recursive left-branching structures such as *[[[Wadeck's Mother's] Friend's] Son]*.[6] It is not clear how we can guarantee that Branch Right sometimes applies not to the rightmost word but to the constituent dominating it.

Though the idea of Branch Right is appealing if we want to resolve the contradiction of direction between derivation and linearization, it has both conceptual and empirical problems. We will try to find other ways to show that the paradox is not a real contradiction.

3. Spell-Out before Merge

---

[5] Kempson et al. (2001) and O'Grady (2005) also propose a similar system of incremental derivation. The arguments against Merge Right to be presented below apply to their derivational systems.

[6] This is the title of an American movie directed by Arnold Barkus (1992).

One way to keep both Merge Up and Linearize Down without contradiction is to assume that Spell-Out sends lexical items with information about the constituent structure to be built before Merge combines two syntactic objects. I will explore this possibility in detail below.

First, I will illustrate the basic idea with the example sentence (1) above. I argue that the computational system puts a lexical item with a syntactic bracket in the working space of derivation and Spells Out them to PF at the same time. Then the derivation proceeds in the following steps:

(16) a. [Alice
b. [Alice [Infl
c. [Alice [Infl [loves
d. [Alice [Infl [loves [small
e. [Alice [Infl [loves [small [hamsters

So far, no Merge applies, but Linearization has applied from top to bottom. In this model, Merge applies when a closing bracket is introduced in the derivation. First, a closing bracket is added to (16e) to make a non-branching constituent [hamsters] as shown in (17).

(17) [Alice [Infl [loves [small [$_{N}$ hamsters]

Then, another closing bracket is introduced into derivation to make the NP *small hamsters* as shown in (18).

(18) [Alice [Infl [loves [$_{NP}$ small [hamsters]]

This process, combining two lexical items, is what is called Merge in the standard assumption. Similarly, a closing bracket is introduced into the derivation to merge two constituents, making a larger constituent, VP, I', and IP, as shown in (19a), (19b), and (19c).

(19) a. [Alice [Infl [$_{VP}$ loves [small [hamsters]]]

b. [Alice [$_{I'}$ Infl [loves [small [hamsters]]]]

c. [$_{IP}$ Alice [Infl [loves [small [hamsters]]]]]

In the derivation shown in (16) to (19), Linearization applies from top to bottom, and Merge applies from bottom to top. Thus, there is no contradiction between Linearize Down and Merge Up in this derivational model. Each lexical item is Spelled Out before being Merged with another syntactic object. There is no phase unit for PF interface such as VP or IP in the sense of Chomsky (2001). Each lexical item is Spelled Out when introduced into the derivation.

Note that Spell-Out strips only phonetic features from lexical items in the working space, as is assumed in the standard theory of derivation and interface. Semantic features remain in the derivation until Merge applies to make semantic units and send semantic representations to LF, perhaps at both the CP and *v*P phases. Syntactic features also remain in the derivation until they are checked and deleted when Merge makes checking domains. In the next section, I will show the details of derivation, the Spell-Out to PF, and semantic features.

4. Spell-Out of brackets as silent beats

A number of questions arise. Are starting brackets and closing brackets syntactic objects? Does the speaker Spell Out brackets? If so, how?

I argue that brackets are real objects in syntactic representation. We need brackets to show that two syntactic objects are merged into one. We may think of brackets as a special

kind of lexical items. They have phonetic features in that they indicate silent pause duration, syntactic features in that they show the boundary of a constituent, and no semantic features. They are included in Numeration with the number of times they appear, and introduced into derivation when they are necessary for convergence. For example, the Numeration of the sentence (1) should be {*Alice, Infl, loves, small, hamsters*, [x5,]x5} for the convergent derivation.

Moreover, it is clear that a sentence has silences or pauses between two words. These silences or pauses should be represented in some form in PF, which must have some corresponding syntactic objects. In other words, Spell-Out sends phonetic features of syntactic objects to PF. PF has some representation for silences or pauses. Then, syntactic representation must have some syntactic objects which are interpreted as PF objects corresponding to silence or pauses. Thus, it is plausible to assume that syntactic brackets are syntactic objects, which have a phonetic (silence) feature and no semantic features.

I also argue that a bracket is introduced into derivation at the same time that a lexical item is introduced. Consider what would happen if no bracket was introduced before the first word, as shown in (20).

(20) Alice

This is not going to give any convergent derivation. Spell-Out strips phonetic features from *Alice* at the step (20). After this Spell-Out, there is no chance for the word to be included in a constituent even if any brackets are introduced, as shown in (21).

(21) *Alice* [loves [small [hamsters]]]]

The derivation (21) crashes at LF because *Alice* is a "stray" in syntactic structure. *Alice* cannot be assigned any thematic role under the standard assumption that theta is assigned by configuration in a syntactic tree (Baker (1988) and Hale and Keyser (1993)). Thus, when a

lexical item is introduced into a derivation, a bracket must also be introduced, as we have shown in (16).

Before considering an example derivation in detail, let us consider what kind of phonological features a bracket has. As I mentioned above, a sentence has pauses and silences between two words. A plausible object to represent pauses and silences in PF is Selkirk's (1984) Silent Demibeat. Silent demibeats are assumed to be contained in the metrical grid of an utterance. Selkirk argues that the representation of the rhythmic structure of the sentence *Abernathy gesticulated* is as follows, where the underscored grids are silent demibeats:

(22)

```
                    x
    x               x
    x   x           x  x
    x x x  x xx x   x x x x  xxx
    Abernathy       gesticulated
```

Selkirk argues that silent demibeats are added at the end (right extreme) of the metrical grid aligned with a word or certain constituents.[7] Here I will assume that a syntactic bracket is

---

[7] Selkirk (1984) argues that the sentence in (ia) contains a number of silent demibeats (x) as shown in (ib).

(i) a. $[_S\ [_{NP}\ [_N$ Mary$]]\ [_{VP}\ [_V$ finished$]\ [_{NP}$ her $[_A$ Russian$]\ [_N$ novel$]]]]$

b. Mary xxx finished xx her Russian x novel xxxxx

a, b, d a, b a a, b, c, d

The silent demibeats in (ib) are assigned by Silent Demibeat Addition (ii).

(ii) Add a silent demibeat at the end (right extreme) of the metrical grid aligned with

a. a word,

b. a word that is the head of a nonadjunct constituent,

c. a phrase,

d. a daughter phrase of S.

Spelled Out as a silent demibeat. We can formulate the idea into the following rule for Spell-Out (cf. Tokizaki 1999):

(23) $\left\{ \begin{matrix} [\\ ] \end{matrix} \right\}$ → x

Spell-Out interprets a syntactic bracket, either left or right (cf. Ferreira 1993, Watson and Gibson 2004), as a silent demibeat. The rule (23), in effect, encodes the syntactic structure (24a) into PF representation (24b).

(24) a. [Alice [loves [small hamsters]]]

b. x Alice x loves x small hamsters xxx

Note that derivation proceeds with multiple Spell-Out. Every time a syntactic object is introduced into derivation, its phonetic features are sent to PF. If it is a syntactic bracket, a silent demibeat is sent to PF.

Now let us consider each step of the derivation in detail. For example, consider the derivation of the sentence (1) again. In each of the derivational steps in (25) below, the first line shows syntactic features, the second semantic features, and the third phonological features.

In (i), according to Selkirk, *Mary* is a word, and a word that is the head of a nonadjunct constituent, and a daughter phrase of S. Thus, three silent demibeats are added to the right of *Mary* by (iia), (iib), and (iic).

Here I will generalize Silent Demibeat Addition. Putting aside (iib) and (iid), a word gets a silent demibeat after it by (iia). If the word is the final one in a phrase, it gets another silent demibeat by (iic). Instead of listing categories receiving a silent demibeat at its right edge as in (iia-d), I assume the syntax-phonology correspondence rule as shown in (23).

(25) a. [Alice

‘Alice’

x ælɪs → PF: x ælɪs

b. [Alice ⟦*Infl*

‘Alice’

PF: x ælɪs

c. [Alice ⟦*Infl* [loves

‘Alice’ ‘loves’

lʌvz → PF: x ælɪs x lʌvz

d. [Alice ⟦*Infl* [loves [small

‘Alice’ ‘loves’ ‘small’

smɔːl → PF: x ælɪs x lʌvz x smɔːl

e. [Alice ⟦*Infl* [loves [small [hamsters

‘Alice’ ‘loves’ ‘small’ ‘hamsters’

x hæmstəˑz → PF: x ælɪs x lʌvz x smɔːl x hæmstəˑz

f. [Alice ⟦*Infl* [loves [small [$_{N}$ hamsters]

‘Alice’ ‘loves’ ‘small’ ‘hamsters’

x → PF: x ælɪs x lʌvz x smɔːl x hæmstəˑz x

g. [Alice ⟦*Infl* [loves [$_{NP}$ small [hamsters]]

‘Alice’ ‘loves’ ‘small’ ‘hamsters’

x → PF: x ælɪs x lʌvz x smɔːl x hæmstəˑz xx

h. [Alice ⟦*Infl* [$_{VP}$ loves [small [hamsters]]]

‘Alice’ ‘loves’ ‘small’ ‘hamsters’

x → PF: x ælɪs x lʌvz x smɔːl x hæmstəˑz xxx

i. [Alice ⟦$_{I'}$ *Infl* [loves [small [hamsters]]]⟧

‘Alice’ ‘loves’ ‘small’ ‘hamsters’ → LF: ‘loves small hamsters’

PF: x æl ɪs x lʌvz x smɔːl x hæmstɚz xxx

j. [$_{IP}$ Alice ⟦*Infl* [loves [small [hamsters]]]⟧*]*

'Alice' → LF: 'Alice loves small hamsters'

x -> PF: x æl ɪs x lʌvz x smɔːl x hæmstɚz xxxx

In (25a), the first lexical item *Alice* and a bracket are introduced into the derivation and their phonetic features are Spelled Out at the same time. The syntactic and semantic features of *Alice* remain in the derivation. In (25b), Infl and a bracket are introduced, but I assume that a phonologically empty element (e.g. Infl and trace) and the brackets merging it with another syntactic object are invisible to Spell-Out (cf. Tokizaki 2006). In (25c), the third item *loves* and a bracket are introduced, and its phonetic features are Spelled Out. What are left in the working space in the derivation are the syntactic and semantic features of *Alice* and *loves*. Similarly, in (25d) and (25e), *small* and *hamsters* are introduced together with a bracket, and are Spelled Out. A first closing bracket is introduced in (25f) and is Spelled Out as a silent demibeat. In (25g) to (25j), a closing bracket allows Merge to make a larger constituent, NP, VP, I', and IP. At a phase (25i), the semantic features of VP are sent to LF. LF is completed at the next phase (25j) where the semantic feature of *Alice* is added.[8]

So far, I have argued that brackets, a kind of syntactic objects with a phonetic pause feature, are Spelled Out as silent demibeats in PF representation. I showed each step of syntactic derivation, the Spell-Out of phonological features to PF, and semantic interpretation.

---

[8] The reviewer raises the question how contraction phenomena such as *wanna* are explained in this system. A possible answer is to assume that contracted forms are listed as items in the lexicon and are introduced into derivation as such, as shown in (i) (cf. Goodall 2006 and the references cited therein).

(i) [I [Infl [wanna [dance [tonight]]]]]

This shows how Linearization proceeds from top to bottom while Merge proceeds from bottom to top.

5. Parsing: Reconstruction of syntax from PF

In this section, I will argue that parser also uses silent demibeats to reconstruct phrase structure. I have argued that Spell-Out interprets a syntactic bracket (either left or right) as a silent demibeat. Here I propose that parser interprets a silent demibeat as a syntactic bracket to reconstruct the intended phrase structure. The interpretation of a silent demibeat might be formulated as follows:

(26) $\underline{x} \rightarrow \left\{ [\;] \right\}$

This rule is the reverse of the Spell-Out rule in (23). However, this time we face the problem of which bracket is the one intended: right or left. Parser has no information about the direction of a bracket.

I propose that a silent demibeat immediately followed by a lexical item is interpreted as a left bracket. This parsing rule can be formulated as (27).

(27) $\underline{x}\ \alpha \rightarrow [\alpha$ (α: a lexical item)

For example, the parsing of the sentence (24b) proceeds as shown in (28). Each step in (28) corresponds to a derivational step in (25).[11]

(28)	PF	Parsing
a.	x ælɪs	*[Alice*
b.	x ælɪs	[Alice ⟦*Infl*
c.	x ælɪs x lʌvz	[Alice ⟦*Infl [loves*
d.	x ælɪs x lʌvz x smɔːl	[Alice ⟦*Infl* [loves *[small*
e.	x ælɪs x lʌvz x smɔːl x hæmstə˞z	[Alice ⟦*Infl* [loves [small *[hamsters*

The question is what happens next. I will use the serial alphabet from (28) to show the correspondence between Spell-Out (25) and the reconstruction.

(29) f. x ælɪs x lʌvz x smɔːl x hæmstə˞z x

The rule in (27) cannot change the silent demibeat following *hæmstə˞z* into a left bracket because the silent demibeat is not followed by a lexical item at this step. Thus, no reconstruction applies until next step shown in (30g).

(30) g. x ælɪs x lʌvz x smɔːl x hæmstə˞z xx

Now we need another parsing rule to change silent demibeats into right brackets. Let us assume the following rule for right brackets:

---

[11] How *Infl* in (28b) is reconstructed from PF is an open question. See the discussion on the invisibility of phonologically empty elements below (25).

(31) xx →]x

Then, parsing proceeds as in (32).[12]

(32)

	PF	Parsing
g.	x ælɪs x lʌvz x smɔːl x hæmstəˑz *xx*	[Alice ⟦*Infl* [loves [small [$_{N}$ hamsters*]*
h.	x ælɪs x lʌvz x smɔːl x hæmstəˑz x*x*x	[Alice ⟦*Infl* [loves [$_{NP}$ small [hamsters]*]*
i.	x ælɪs x lʌvz x smɔːl x hæmstəˑz xx*x*	[Alice ⟦*Infl* [loves [small [hamsters]]
j.	x ælɪs x lʌvz x smɔːl x hæmstəˑz xx*xx*	[Alice ⟦$_{I'}$ *Infl* [$_{VP}$ loves [small [hamsters]]*]]*
k.	x ælɪs x lʌvz x smɔːl x hæmstəˑz xxx*xx*	[$_{IP}$ Alice ⟦*Infl* [loves [small [hamsters]]]⟧*]*

Thus, we can explain how parser builds a syntactic tree from pause durations between words.[13]

Notice that in (32), as well as the structures we have seen in section 3 and section 4, only the rightmost lexical item *hamsters* by itself is contained in a pair of brackets. In a tree diagram, the lexical item would be dominated by a non-branching node.

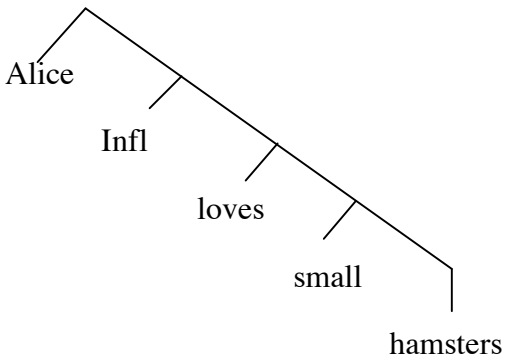
---

[12] In fact, we need one more silent demibeat as shown in (32k) than we have seen in (25j) in order to Merge *Alice* and the I'. I argue that the final silent demibeat is supplied by the next sentence in a discourse. The PF in (32k) is mapped from such discourse as (i), which has six brackets between *hamsters* and *she*.

(i) [Alice *[Infl* [loves [small [hamsters]]]*]*] [She ....

[13] How the right bracket of I' is reconstructed from PF in (32j) is an open question. See note 10.

(33)



Interestingly, this tree is the same as Kayne's (1994) tree based on the Linear Correspondence Axiom. Kayne argues that syntactic structure must be asymmetric and that the rightmost element in a tree must be immediately dominated by a non-branching node. For example, Kayne (1994:10) shows the following structure, where N is immediately dominated by a non-branching NP node.

(34) $[_K$ J $[_{VP}$ $[_V$ see] $[_{NP}$ $[_N$ John]]]]

The analysis of linearization and parsing presented here also needs the rightmost non-branching node in a tree, as we have seen in section 4 and this section. The similarity between Kayne's idea of asymmetry and the analysis presented here may be rooted in a deep principle of language, but I will not pursue it here.

6. Derivation and parsing of left-branching structure

Now let us turn to cases of left-branching structure. For example, consider the following sentence:

(35) [[Alice [Walker]] [Infl [loves [hamsters]]]]

Here the subject NP branches into two words. Let us focus on the initial part of the derivation, Spell-Out, and parsing of (35).

(36)		syntax	PF	Parsing
	a.	[[Alice	xx ælɪs	]x Alice
	b.			] [Alice

Here, two brackets are introduced into the derivation with *Alice* in order to get the convergent derivation. The Spell-Out rule (23) applies to (35) to give a PF with silent demibeats. However, the parsing rule (31) wrongly changes the first silent demibeat to a right bracket at the first step (36a). However, the parsing in (36b) is a vacuous closing of a constituent because there is no lexical item to be enclosed by the right bracket and no left bracket to be paired with. Such a vacuous parsing of a silent demibeat should be banned in principle. Then, in this case, parser must interpret the first silent demibeat as a left bracket as shown in (37).

(37)		syntax		PF		Parsing
	a.	[[Alice	→	xx ælɪs	→	[x Alice
	b.					[[Alice

It has been pointed out that in right-branching languages such as English, left-branching structure is somewhat marked compared to right-branching structure (cf. Quirk et al. 1985). We can argue that this markedness of left-branching structure may come from the marked interpretation of double silent demibeats.

The derivation, Spell-Out, and parsing following (37) are straightforward as shown in (38), where I omit *Infl* for the sake of simplicity of illustration.

(38) a. [[Alice *[Walker* → *x* wɔːkəʳ → [[Alice *[Walker*

b. [[Alice [Walker*]]* → wɔːkəʳ *x*x → [[Alice [Walker*]*

c. [[Alice [Walker]] *[loves* → wɔːkəʳ *xxx* loves → [[Alice [Walker]*] [*loves

The rest is similar to (25d) to (25j).

Thus, we conclude that left-branching structure can also be derived by the Spell-Out and Merge operation we have proposed for right-branching structure.[14] For parsing, I argued that left-branching structure can be parsed by marked interpretation of silent demibeats as shown in (37a), but not by the unmarked interpretation (31).

7. Phonological evidence for the analysis

Finally, let us look at some phonological evidence for the analysis presented above. First, let us consider the prosody of structurally ambiguous sentences. Cooper and Paccia-Cooper (1980) report that speakers put a longer pause between *cop* and *with* in (39a) than in (39b).

(39) a. Jeffrey hit [the cop] [with a stick] [127.7 msec] (Jeffrey had the stick)

b. Jeffrey hit [the cop_with a stick] [97.1 msec] (The cop had the stick)

The Spell-Out and Merge operation we have argued derives the following syntactic structures:

(40) a. [Jeffrey [hit [the [cop]] [with [a [stick]]]]]

b. [Jeffrey [hit [the [cop [with [a [stick]]]]]]]

[14] For the possibility of different juncture between left-branching and right-branching structure, see Tokizaki (2008).

The brackets are Spelled Out as silent demibeats by the rule (23) as shown in (41).

(41) a. x dʒefri x hɪt x ðə x kɑp xxx wɪð x ə x stɪk xxxxx

b. x dʒefri x hɪt x ðə x kɑp x wɪð x ə x stɪk xxxxxxx

The number of silent demibeats between *cop* and *with* is three in (41a) and one in (41b). A silent demibeat is interpreted as a certain length of pause at the articulatory-perceptual system (A-P). Then, we correctly expect a longer pause for the three silent demibeats between *cop* and *with* in (41a) than for the one silent demibeat at that position in (41b). The Spell-Out and Merge operation proposed here explains why the pause duration at that point is longer in (39a) than in (39b). Thus, the different pause lengths in structurally ambiguous sentences give support to the analysis presented here.

For parsing, we also have some evidence for our analysis. It has been argued that hearers tend to interpret a prosodic break as a major constituent break. For example, according to Pynte and Prieur (1996), if the second prosodic break is put between NP and PP in (42), the interpretation (43b) is preferred to (43a).

(42) The spies [$_{VP}$ informed # [$_{NP}$ the guards] (#) [$_{PP}$ of NP]]

(43) a. The spies [informed [the [guards [of [the palace]]]]]

b. The spies [informed [the [guards]] [of [the conspiracy]]]]

We can explain this preference with the analysis presented above. I argue that a prosodic break in the A-P system corresponds to a certain number of silent demibeats in PF representation (cf. Tokizaki 2006). Then, it is plausible to assume that the second prosodic break between NP and PP is perceived by a hearer as two more silent demibeats as shown in (44b).

(44)	A-P	PF	Parsing	
a.	.. gɑːdz əv ..	.. gɑːdz x əv ..	.. guards [of ..	(= (43a))
b.	.. gɑːdz # əv ..	.. gɑːdz *x*x	.. guards*]*	
		.. gɑːdz x*x*x	.. guards]*]*	
		.. gɑːdz xx*x* əv ..	.. guards]] *[*of ..	(= (43b))

A hearer interprets a prosodic break between *guards* and *of* as a series of silent demibeats as shown in (44b) rather than (44a). The silent demibeats are parsed by (31) and then by (27) as two right brackets and a left bracket, as shown in the steps in (44b). Thus, we explain why a prosodic break tends to be interpreted as a major constituent break in cases such as (42). This discussion gives support to the parsing system proposed above.

In this section, I have argued that some production and perception data can be naturally explained with the analysis presented here. This fact gives support for the Spell-Out before Merge hypothesis.

8. Summary

I have argued that we can resolve the contradiction in direction between top-down linearization and bottom-up tree structuring if we assume a Spell Out before Merge hypothesis. Computational system Spells Out a lexical item and a bracket to PF stepwise as a sound and a silent demibeat. Merge is triggered by inserting a right bracket, which encloses a constituent with a preceding left bracket. Parser interprets the silent demibeats in PF representation as syntactic brackets which build a syntactic tree. Thus, there is no paradox between Linearize Down and Merge Up. I hope that the analysis of the syntax-PF interface presented here will constitute a small but steady step toward the overall goal of the minimalist program.

References

Baker, M. 1988. *Incorporation*. Chicago: Chicago University Press.

Chomsky, N. 1995. *The minimalist program*. MIT Press.

Chomsky, N. 2001. Derivation by phase. *Ken Hale: A life in language*, ed. by M. Kenstovicz, 1-52. MIT Press.

Chomsky, N. and M. Halle. 1968. *The sound pattern of English*. New York: Harper and Row.

Collins, C. 2002. Eliminating labels. *Derivation and explanation in the minimalist program*, ed. Samuel David Epstein and T. Daniel Seely, 42-64. Blackwell.

Cooper, E. and J. Paccia-Cooper. 1980. *Syntax and speech*. Harvard Univ. Press.

Dobashi, Y. 2003. *Phonological phrasing and syntactic derivation*. Doctoral dissertation. Cornell University.

Epstein, S. D. et al. 1998. *A derivational approach to syntactic relations*. Oxford University Press.

Ferreira, F. 1993. Creation of prosody in sentence production. *Psychological Review* 100, 233-253.

Goodall, Grant. 2006. Contraction. *The Blackwell companion to syntax: vol. 1*, eds. Martin Everaert and Henk van Riemsdijk, 688-703. Oxford: Blackwell.

Hale, K. and S. J. Keyser. 1993. On argument structure and the lexical expression of semantic relations. In *The View from Building 20*, ed. K. Hale and S. J. Keyser. Cambridge, Mass.: MIT Press, 53-109.

Hawkins, J. 1994. *A performance theory of order and constituency*. Cambridge University Press.

Kayne, R. 1994. *The antisymmetry in syntax*. MIT Press.

Kempson, Ruth, Wilfried Meyer-Viol, and Dov Gabbay. 2001. *Dynamic syntax: The flow of language understanding*. Malden, MA: Blackwell.

Larson, R. 1990. Double objects revisited: Reply to Jackendoff, *Linguistic Inquiry* 21, 589-632.

Nespor, M., and M. Scorretti. 1984. Empty elements and phonological form. In *Grammatical representation*, ed. Jacqueline Guéron, Hans-Georg Obenauer and Jean-Yves Pollock, 223-235. Dordrecht: Foris.

O'Graday, W. 2005. *Syntactic carpentry: An emergentist approach to syntax*. Lawrence Erlbaum Associates.

Phillips, C. 1996. *Order and structure*. Doctoral dissertation, MIT.

Phillips, C. 2003. Linear order and constituency. *Linguistic Inquiry* 34, 37-90.

Pynte, J. and B. Prieur. 1996. Prosodic breaks and attachment decisions in sentence parsing. *Language and Cognitive Process* 11: 165-191.

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik. 1985. *A comprehensive grammar of the English language*. London: Longman.

Richards, N. 1999. Dependency formation and directionality of tree construction. *MIT Working Papers in Linguistics* 34, 67-105.

Selkirk, E. 1984. *Phonology and syntax*. MIT Press.

Shiobara, K. 2005. *Linearization: A derivational approach to the syntax-prosody interface*. Doctoral dissertation, The University of British Columbia.

Tokizaki, H. 1999. Prosodic phrasing and bare phrase structure. *NELS* 29, vol. 1, 381-395.

Tokizaki, H. 2005. Prosody and phrase structure without labels, *English Linguistics* 22, 380-405.

Tokizaki, H. 2006. *Linearizing syntactic structure with brackets and silence: A minimalist theory of syntax-phonology interface*. Doctoral dissertaion, the University of Tsukuba. A revised version is published as *Syntactic structure and silence*, Tokyo: Hitsuji Shobo, 2008.

Tokizaki, H. 2008. Symmetry and asymmetry in the syntax-phonology interface. *Phonological Studies* 11.

Uriagereka, J. 1998. *Rhyme and reason: An introduction to minimalist syntax*, Cambridge, MA, MIT Press.

Uriagereka, J. 1999. Multiple Spell Out. *Working minimalism,* ed. by Samuel David Epstein and Norbert Hornstein, 251-281. Cambridge, Mass. : MIT Press.

Wasow, T. 2002. *Postverbal behavior*. CSLI.

Watson, D. and E. Gibson. 2004. The relationship between intonational phrasing and syntactic structure in language production. *Language and Cognitive Processes* 19, 713-755.

Department of English

Sapporo University

Nishioka 3-7

Sapporo, Japan 062-8520

toki@sapporo-u.ac.jp